TotalView

New Features

ETNUS

April 2003

Version 6.2

Etnus has prepared this manual for the exclusive use of its customers, personnel, and licensees. The information in this manual is subject to change without notice, and should not be construed as a commitment by Etnus. Etnus assumes no responsibility for any errors that appear in this document.

Contents

New Features

New Features

This booklet contains information about changes from TotalView® version 5 to TotalView® version 6, 6.1, and 6.2. Items that are unique to version 6.1 and 6.2 are noted.

Behind the Scenes

Symbol Table Reworking

When you look at information in TotalView, you'll see some apparently innocuous changes in the way information is described. For example, you might see a variable described as follows in a Variable Window:

```
arrays.F#MAIN#master_array
```

In the Stack Frame Pane, you'll often see variables separated by headers saying something like **Block $b2$b3**.

These descriptions indicate a heightened awareness of where blocks are declared and what their scope is. Something like **$b2$b3** indicates that a variable exists within a routine's second unnamed block and that unnamed block is also an unnamed block.

While this change appears rather simple, it is reflected in almost every thing you do and see.

For more information, see *Qualifying Symbol Names*.

Non-Uniform Address Space

When debugging a program running on more than one computer, TotalView has required that each computer be running the same version of the operating system. In addition, the libraries that exist on each of these computers had to be the same. This restriction has been relaxed. While the program must be running on the same kind of computer, the operating system and its libraries can be any of these supported by this version of TotalView.

Performance Improvements (version 6.1)

TotalView's performance when attaching to processes in large multi-processor jobs has been improved.

UPC is Now Supported (version 6.1)

UPC is supported on gcc running on SGI IRIX—this is also called Intrepid—and on HP Tru64 systems.

CodeRoad

The CodeRoad JNI Bridge lets you seamlessly debug your Java applications that use the Java Native Interface (JNI) by easily moving from your Java code to your C/C++ code and back. The JNI Bridge allow you to perform non-stop debugging across the two environments, giving you a full and clear picture of your application's execution path.

CodeRoad is purchased separately from TotalView.

New and Changed Commands

General Changes

- **Type Transformation Facility (version 6.2)**: When you have enabled this facility, TotalView cleans up the display of STL variables. You can turn this facility on and off by selecting **View simplified STL containers (and user-defined transformations)** in the **Options** Page of the **File > Preferences** command.

The CREATING TYPE TRANSFORMATION GUIDE explains how you can write your own transformations.

- **Process Window's width and group controls simplified (version 6.1)**: The dual pulldowns on the toolbar have been combined into a simplified and unified control.
- **Middle mouse diving**: You can now perform dive operations using your middle mouse button. Double-clicking with your left mouse button is still supported.
- **Forward-backward icons**: After diving in the Source Pane or in the Variable Window, you've always been able to use the "<" icon to go back to where you came from. To the right of this icon is a ">" icon that redives back to a the information you "undived" from.
- **Data precision**: You can now tell TotalView the precision at which it should display numbers and strings. This is set using the Precision Page from the **File > Preferences** window.

  For more information, see the **File > Preferences** help information.
- **Dereferencing**: You can tell TotalView to automatically dereference information and change information into arrays. This is set using the Pointer Dive Page within the **File > Preferences** Window.

  For more information, see the **File > Preferences** command.

New GUI Commands

- (Process and Variable Windows) **View > Redive**: After you perform an undive operation, this command restores the display to what it was before the undive operation.
- (Variable Window) **View > Dive in All** and **View > Dive in All Anew**: These two commands lets you view a single element in an array of structures as a single array. The "Anew" version displays the data in a different window instead of just replacing the Variable Window's contents.

  For more information, see "*Displaying Array of Structure Elements*. in the TOTALVIEW USERS GUIDE"
- (Process Window) **Group > Edit Group**: Using this window, you can define and change P/T (Process/Thread) Set group membership.

  For more information, see "*Using the Group Editor*".
- (Process Window) **Tools > Program Browser**: The **Global Variables** window is replaced by the Program Browser After you display a list of programs and libraries, dive on the library to see the object's variables.

  For more information, see "*Browsing for Variables*".

- (Process Window) **Memorizing positions**: Using commands on the Windows menu, you can have TotalView remember where you've placed you windows so that they come up on the same place the next time you invoke the window.

  For more information, see "*Resizing and Positioning Windows and Dialog Boxes*."
- (Process Window) **Tools > Memory Usage**: Displays information about how much memory the processes within your program are using. The use is shown by process areas such as text, data, heap, and stack.

  For more information, see "*Displaying Memory Statistics*."
- (Root Window) **Tools > P/T Set Browser**: Using this window, you can see what processes and threads are members of a group. You can also explore which processes and threads will become members of a group.

  For more information, see *Using the P/T Set Browser*.
- (Variable Window) **View > Load Symbols**: When using the **Tools > Program Browser** command to display your program's symbols, TotalView does not display symbols created by your loader. If you need to see these symbols, use this command.
- (Variable Window): **Tools > Visualization Distribution**: When visualizing distributed arrays, selecting this command tells the Visualizer that one of the dimensions in the displayed information should be the node associated with the variable.

Changed GUI Commands

- **Action Point > Properties**: TotalView's ability to set action points has been enhanced. In previous versions, setting action points on some kinds of statements required TotalView to present a list of all possible places, and then ask you which ones you wanted to set them at. In version 6, TotalView usually does the right thing so you are not asked what to do. Here are examples of places where things happen more automatically: template instances, inline functions, multiple statements on a line, optimized code, and loops.

  If what occurs is not what you want to occur, you can open the Properties dialog box and, by selecting the **Address** button, make changes to what is automatically set.

  When you are setting a breakpoint, the **Processes** button lets you select the enabled/disabled state of all processes that would participate in the share group. Previously, you had to go to the process to do this.
- **Edit > Find**: The Find dialog box stays up unless you unselect the Keep Dialog checkbox. If the text you're looking for isn't found, you can have Find continue searching from the beginning of the file.

- Tools > Call Tree: This window now has P/T set controls. This means that you can specify which processes and threads display information.
- Tool > Evaluate Window: This window now has P/T set controls. This lets you evaluate code across processes and threads.
- **Viewing interleaved assembler**: The View > Assembler > Interleaved command is now the View > Assembler > Both command.
- Message Queue Graph: This window now has P/T set controls. This lets you indicate which processes and threads display message queue information.

New CLI Commands

- dcache: Clears the remote library cache. If you are running a multi-process program running on another computer, TotalView may store library information from these remote computers in its **.totalview** subdirectory. You may want to delete this cache if disk space is limited.

  For more information, see the dcache command.
- dflush: Unwinds the stack created by **dprint** computations. When dprint is evaluating multiple functions, it is possible that your program can be suspended. By flushing the stack, you unblock your program.

  For more information, see the dflush command.
- dmstat: Displays memory use information.

  For more information, see the dmstat</a> command.
- TV::expr: Manipulates values created by dprint **-nowait**.

  For more information, see the TV::expr command.

Changed CLI Commands

- capture: You can now redirect I/O.

  For more information, see the capture command.
- dattach, **–c option**: You can now attach to a core file.

  For more information, see the dattach command.
- dprint, **calling functions**: Calling functions from the dprint command has been greatly improved. Because evaluation can occur across process, the **-nowait** option allows processing to occur asynchronously. to obtain these results, you'll need to use the TV::expr command.

  For more information, see the dprint command.
- dwhere **now uses the** MAX_LEVELS **variable**: Using this variable, you can limit how many levels are displayed when you use this command.

  For more information, see the dwhere command.

Other Changes

- **Entering Environment Variables (6.2):** In previous versions, entering an environment variable deleted environment variables passed in to TotalView when you invoked TotalView from a shell. At this release, TotalView variables no longer overwrites these variables when you add a new variable using the **Process > Startup Parameters** command. Instead, the variable you enter is added to your environment.
- **.totalview directory**: TotalView preferences, startup file, and its cache are stored within a **.totalview** subdirectory contained within your home directory.

  For more information, see "*Initializing TotalView*".
- **Xdefaults, CLI variables, and command-line options**: Most of the TotalView states that could only be set using and **.Xdefaults** file can now be set using CLI variables. (These variables are in the **TV::** and **TV::GUI::** namespaces.) Consequently, the only states placed in an **.Xdefaults** file are those indicated on our web site. See http://www.etnus.com/Support/docs for more information. While these **.Xdefaults** will continue to work at release 6, they may not work in future releases
- **-e option to totalview and totalviewcli commands**: Tells TotalView to immediately execute the CLI commands that are the argument to this command.
- **Documentation**: The TotalView documentation has been reorganized. All reference information that was contained in the release 5.0 totalView users Guide and TotalView CLI Guide is now contained within the TotalView Reference Guide. The remaining information is in the TotalView User Guide. The Converter Guide, which described the differences between releases 4 and 5 will no longer be available as part of the documentation set. It will be available on our web site.